"Express Mail" mailing label number EV 164034146 US

Date of Deposit: December 02, 2003

Attorney Docket No.15147US02

SYSTEM, METHOD, AND APPARATUS FOR DISPLAY MANAGER

RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application, Serial No. 60/516,387, filed October 31, 2003, entitled "System, Method, and Apparatus for Display Manager" (Attorney Docket 15147US01), by Savekar, et. al., which is incorporated herein by reference.

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0002] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[0003] [Not Applicable]

BACKGROUND OF THE INVENTION

[0004] Video decoders decode a video bit-stream encoded according to a predetermined standard syntax, such as MPEG-2 or Advanced Video Compression (AVC). An encoder generating a compressed video bit-stream makes a number of choices for converting the video stream into a compressed video bit-stream that satisfies the quality of service and bit-rate requirements of a channel and media. However, decoders have limited choices while decoding the compressed bit stream. The decoder uses the decisions made by the encoder to decode and present pictures at the output screen with the correct frame rate at the correct times, and the correct spatial resolution.

[0005] Decoding can be partitioned in two processes - the decode process and the display process. The decode

process parses through the incoming bit stream and decodes the bit stream to produce decode images which contain raw pixel data. The display process displays the decoded images onto an output screen at the proper time and at the correct and appropriate spatial and temporal resolutions as indicated in the display parameters received with the stream.

[0006] The decoding and display process are usually implemented as firmware in SRAM executed by a processor. The processor is often customized and proprietary, and embedded. This is advantageous because the decoding process and many parts of the displaying process are very hardware-dependent. A customized and proprietary processor alleviates many of the constraints imposed by an off-the-shelf processor. Additionally, the decoding process is computationally intense. The speed afforded by a customized proprietary processor executing instructions from SRAM is a tremendous advantage. The drawbacks of using a customized proprietary processor and SRAM are that the SRAM is expensive and occupies a large area in an integrated circuit. Additionally, the use of proprietary and customized processor complicates debugging. The software for selecting the appropriate frame for display has been found, empirically, to be one of the most error-prone processes. Debugging of firmware for a customized and proprietary processor is complicated because few debugging tools are likely to exist, as compared to an off-the-shelf processor.

[0007] Further limitations and disadvantages of conventional and traditional approaches will become apparent to one of ordinary skill in the art through comparison of such systems with the present invention as set forth in the

remainder of the present application with reference to the drawings.

BRIEF SUMMARY OF THE INVENTION

[0008] Aspects of the present invention may be seen in a method for displaying images using a circuit in a system that comprises a decoder for decoding encoded images and parameters associated with the images; image buffers for storing the decoded images; parameter buffers for storing the decoded parameters associated with the decoded images; a display engine for receiving the decoded parameters and displaying the decoded images based on the decoded parameters; and a display manager for determining the display order of the decoded images. The system further comprises a first processor and a second processor, and a first memory and a second memory.

[0009] The circuit comprises a decoder; image buffers connected to the decoder and configured to store images decoded by the decoder; parameter buffers connected to the decoder and configured to store parameters associated with the images and decoded by the decoder; a display engine connected to the image buffers and the parameter buffers and configured to receive the decoded parameters from the parameter buffers and display the decoded images based on the decoded parameters; and a display manager connected to the display engine and configured to determine the display order for the decoded images based on the decoded parameters. The circuit further comprises a first processor and second processor, and a first memory and a second memory.

[0010] The method for displaying images comprises decoding the images; decoding parameters associated with the images; buffering the decoded images; buffering the decoded parameters associated with the decoded images; determining the display order for the decoded images based on the

associated decode parameters; and displaying the decoded images based on the associated decoded parameters and based on the determined display order.

[0011] These and other features and advantages of the present invention may be appreciated from a review of the following detailed description of the present invention, along with the accompanying figures in which like reference numerals refer to like parts throughout.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0012] Fig. 1a illustrates a block diagram of an exemplary Moving Picture Experts Group (MPEG) encoding process, in accordance with an embodiment of the present invention.

[0013] Fig. 1b illustrates an exemplary sequence of frames in display order, in accordance with an embodiment of the present invention.

[0014] Fig. 1c illustrates an exemplary sequence of frames in decode order, in accordance with an embodiment of the present invention.

[0015] Fig. 2 illustrates a block diagram of an exemplary circuit for decoding the compressed video data, in accordance with an embodiment of the present invention.

[0016] Fig. 3 illustrates a block diagram of an exemplary decoder and display engine unit for decoding and displaying video data, in accordance with an embodiment of the present invention.

[0017] Fig. 4 illustrates a dynamic random access memory (DRAM) unit 309, in accordance with an embodiment of the present invention.

[0018] Fig. 5 illustrates a timing diagram of the decoding and displaying process, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0019] **Fig.** 1a illustrates a block diagram of an exemplary Moving Picture Experts Group (MPEG) encoding process of video data 101, in accordance with an embodiment of the present invention. The video data 101 comprises a series of frames 103. Each frame 103 comprises two-dimensional grids of luminance Y, 105, chrominance red Cr, 107, and chrominance blue C_b, 109, pixels. The two-dimensional grids are divided into 8x8 blocks, where a group of four blocks or a 16x16 block 113 of luminance pixels Y is associated with a block 115 of chrominance red C_r, and a block 117 of chrominance blue C_b pixels. The block 113 of luminance pixels Y, along with its corresponding block 115 of chrominance red pixels C_r, and block 117 of chrominance blue pixels C_b form a data structure known as a macroblock 111. The macroblock 111 also includes additional parameters, including motion vectors, explained hereinafter. Each macroblock 111 represents image data in a 16x16 block area of the image.

[0020] The data in the macroblocks 111 is compressed in accordance with algorithms that take advantage of temporal and spatial redundancies. For example, in a motion picture, neighboring frames 103 usually have many similarities. Motion causes an increase in the differences between frames, the difference being between corresponding pixels of the frames, which necessitate utilizing large values for the transformation from one frame to another. The differences between the frames may be reduced using motion compensation, such that the transformation from frame to frame is minimized. The idea of motion compensation is based on the fact that when an object moves across a screen, the object may appear in different positions in different frames, but the object itself does not change substantially in

appearance, in the sense that the pixels comprising the object have very close values, if not the same, regardless of their position within the frame. Measuring and recording the motion as a vector can reduce the picture differences. The vector can be used during decoding to shift a macroblock 111 of one frame to the appropriate part of another frame, thus creating movement of the object. Hence, instead of encoding the new value for each pixel, a block of pixels can be grouped, and the motion vector, which determines the position of that block of pixels in another frame, is encoded.

[0021] Accordingly, most of the macroblocks 111 are compared to portions of other frames 103 (reference frames). When an appropriate (most similar, i.e. containing the same object(s)) portion of a reference frame 103 is found, the differences between the portion of the reference frame 103 and the macroblock 111 are encoded. The location of the portion in the reference frame 103 is recorded as a motion vector. The encoded difference and the motion vector form part of the data structure encoding the macroblock 111. In the MPEG-2 standard, the macroblocks 111 from one frame 103 (a predicted frame) are limited to prediction from portions of no more than two reference frames 103. It is noted that frames 103 used as a reference frame for a predicted frame 103 can be a predicted frame 103 from another reference frame 103.

[0022] The macroblocks 111 representing a frame are grouped into different slice groups 119. The slice group 119 includes the macroblocks 111, as well as additional parameters describing the slice group. Each of the slice groups 119 forming the frame form the data portion of a picture structure 121. The picture 121 includes the slice

groups 119 as well as additional parameters that further define the picture 121.

[0023] The parameters may include, for example, a picture structure indicator (frame/top-field/bottom-field), a progressive picture sequence flag (usually comes in transport layer), a progressive frame flag, pan-scan vectors, an aspect ratio, a decode and display horizontal size parameter, a decode and display vertical size parameter, a top field first parameter, and a repeat first field parameter. It is noted that in varying standards there may be additional or less parameters.

[0024] Other parameters may also be functions of defined parameters. For example, the Still Picture Interpolation Mode (SPIM) is a function of the picture structure indicator and the progressive frame/progressive sequence flag. The SPIM represents the display interpolation mode to be used for a still picture and Personal Video Recording (PVR) application such as slow motion when real time decode is turned off. The SPIM controls the way a static frame picture can be displayed onto a screen, for example when a user wishes to pause on a certain frame or when the encoders encode the presentation time stamps of pictures in stream such that decoders are forced to display one frame repetitively. These actions can include displaying the last field, displaying the last displayed top and bottom field pair alternatively, and down-converting the entire frame lines to either top-field or bottom field. The amount of motion between two fields of a frame determines which SPIM mode gives the best visual quality.

[0025] Another example, the motion picture interpolation mode (MPIM) is also a function of the picture structure indicator, progressive frame flag, and progressive sequence flag. The MPIM is a one-bit value

used while displaying moving pictures. If the bit is set, then a complete progressive frame is output onto the screen instead of breaking it into top and bottom fields. If the bit is reset, then the top or bottom field is sent depending on if the display hardware requires the top or the bottom field.

[0026] The progressive frame parameter indicates whether the picture has been encoded as a progressive frame. If the bit is set, the picture has been encoded as a progressive frame. If the bit is not set, the picture has been encoded as an interlaced frame.

[0027] The picture structure parameter specifies the picture structure corresponding to the image buffer. Pan scan vectors specify the displayable part of the picture. The aspect ratio indicates the aspect ratio of the image buffer. The decode and display horizontal size parameters indicate the decoded and the displayable horizontal sizes of the image buffer, respectively.

[0028] The top field first parameter is a one-bit parameter that indicates for an interlaced sequence whether the top field should be displayed first or the bottom field should be displayed first. When set, the top field is displayed first, while when cleared, the bottom field is displayed first.

[0029] The repeat first field is a one-bit parameter that specifies whether the first displayed field of the picture is to be redisplayed after the second field, for an interlaced sequence. For progressive sequence, the repeat first field forms a two-bit binary number along with the top field first parameter specifying the number of times that a progressive frame should be displayed.

[0030] I_0, B_1, B_2, P_3, B_4, B_5, and P_6, **Fig.** 1b, are exemplary pictures representing frames. The arrows

illustrate the temporal prediction dependence of each picture. For example, picture B_2 is dependent on reference pictures I_0, and P_3. Pictures coded using temporal redundancy with respect to exclusively earlier pictures of the video sequence are known as predicted pictures (or P-pictures), for example picture P_3 is coded using reference picture I_0. Pictures coded using temporal redundancy with respect to earlier and/or later pictures of the video sequence are known as bi-directional pictures (or B-pictures), for example, pictures B_1 is coded using pictures I_0 and P_3. Pictures not coded using temporal redundancy are known as I-pictures, for example I_0. In the MPEG-2 standard, I-pictures and P-pictures are also referred to as reference pictures.

[0031] The foregoing data dependency among the pictures requires decoding of certain pictures prior to others. Additionally, the use of later pictures as reference pictures for previous pictures requires that the later picture be decoded prior to the previous picture. As a result, the pictures cannot be decoded in temporal display order, i.e. the pictures may be decoded in a different order than the order in which they will be displayed on the screen. Accordingly, the pictures are transmitted in data dependent order, and the decoder reorders the pictures for presentation after decoding. I_0, P_3, B_1, B_2, P_6, B_4, B_5, **Fig.** 1c, represent the pictures in data dependent and decoding order, different from the display order seen in **Fig.** 1b.

[0032] The pictures are then grouped together as a group of pictures (GOP) 123. The GOP 123 also includes additional parameters further describing the GOP. Groups of pictures 123 are then stored, forming what is known as a video elementary stream (VES) 125. The VES 125 is then packetized to form a packetized elementary sequence. Each packet is

then associated with a transport header, forming what are known as transport packets.

[0033] The transport packets can be multiplexed with other transport packets carrying other content, such as another video elementary stream 125 or an audio elementary stream. The multiplexed transport packets form what is known as a transport stream. The transport stream is transmitted over a communication medium for decoding and displaying.

[0034] **Fig.** 2 illustrates a block diagram of an exemplary circuit for decoding the compressed video data, in accordance with an embodiment of the present invention. Data is received and stored in a presentation buffer 203 within a Synchronous Dynamic Random Access Memory (SDRAM) 201. The data can be received from either a communication channel or from a local memory, such as, for example, a hard disc or a DVD.

[0035] The data output from the presentation buffer 203 is then passed to a data transport processor 205. The data transport processor 205 demultiplexes the transport stream into packetized elementary stream constituents, and passes the audio transport stream to an audio decoder 215 and the video transport stream to a video transport processor 207 and then to a MPEG video decoder 209. The audio data is then sent to the output blocks, and the video is sent to a display engine 211.

[0036] The display engine 211 scales the video picture, renders the graphics, and constructs the complete display. Once the display is ready to be presented, it is passed to a video encoder 213 where it is converted to analog video using an internal digital to analog converter (DAC). The digital audio is converted to analog in an audio digital to analog converter (DAC) 217.

[0037] The decoder 209 decodes at least one picture, I_0, B_1, B_2, P_3, B_4, B_5, P_6...during each frame display period, in the absence of PVR modes when live decoding is turned on. Due to the presence of the B-pictures, B_1, B_2, the decoder 209 decodes the pictures, I_0, B_1, B_2, P_3, B_4, B_5, P_6... in an order that is different from the display order. The decoder 209 decodes each of the reference pictures, e.g., I_0, P_3, prior to each picture that is predicted from the reference picture. For example, the decoder 209 decodes I_0, B_1, B_2, P_3, in the order, I_0, P_3, B_1, and B_2. After decoding I_0 and P_3, the decoder 209 applies the offsets and displacements stored in B_1 and B_2, to the decoded I_0 and P_3, to decode B_1 and B_2. In order to apply the offset contained in B_1 and B_2, to the decoded I_0 and P_3, the decoder 209 stores decoded I_0 and P_3 in memory known as frame buffers 219. The display engine 211, then displays the decoded images onto a display device, e.g. monitor, television screen, etc., at the proper time and at the correct spatial and temporal resolution.

[0038] Since the images are not decoded in the same order in which they are displayed, the display engine 211 lags behind the decoder 209 by a delay time. In some cases the delay time may be constant. Accordingly, the decoded images are buffered in frame buffers 219 so that the display engine 211 displays them at the appropriate time. Accomplishing a correct display time and order, the display engine 211 uses various parameters decoded by the decoder 209 and stored in the parameter buffer 221, also referred to as Buffer Descriptor Structure (BDS).

[0039] **Fig.** 3 illustrates a block diagram of an exemplary decoder and display engine unit for decoding and displaying video data, in accordance with an embodiment of the present invention. The decoder and display engine work together to decode and display the video data. Part of the decoding and

displaying involves determining the display order of the decoded frames utilizing the parameters stored in parameter buffers.

[0040] A conventional system may utilize one processor to implement the decoder 209 and display engine 211. The decoding and display process are usually implemented as firmware in SRAM executed by a processor. The processor is often customized and proprietary, and embedded. This is advantageous because the decoding process and many parts of the displaying process are very hardware-dependent. A customized and proprietary processor alleviates many of the constraints imposed by an off-the-shelf processor. Additionally, the decoding process is computationally intense. The speed afforded by a customized proprietary processor executing instructions from SRAM is a tremendous advantage. The drawbacks of using a customized proprietary processor and SRAM are that the SRAM is expensive and occupies a large area in an integrated circuit. Additionally, the use of proprietary and customized processor complicates debugging. The software for selecting the appropriate frame for display has been found, empirically, to be one of the most error-prone processes. Debugging of firmware for a customized and proprietary processor is complicated because few debugging tools are likely to exist, as compared to an off-the-shelf processor.

[0041] The functionality of the decoder and display unit can be divided into three functions. One of the functions can be decoding the frames, another function can be displaying the frames, and another function can be determining the order in which a decoded frame shall be displayed.

[0042] Referring now to **Fig.** 3, there is illustrated a block diagram of the decoder system in accordance with an

embodiment of the present invention. The second processor 307 oversees the process of selecting a decoded frame from the DRAM 309 for display and notifies the first processor 305 of the selected frame. The second processor 307 executes code that is also stored in the DRAM 309. The second processor 307 can comprise an "off-the-shelf" processor, such as a MIPS or RISC processor. The DRAM 309 and the second processor 307 can be off-chip. The system comprises a processor 305, a memory unit (SRAM) 303, a processor 307, and a memory unit (DRAM) 309.

[0043] The first processor 305 oversees the process of decoding the frames of the video frames, and displaying the video images on a display device 311. The first processor 305 may run code that may be stored in the SRAM 303. The first processor 305 and the SRAM 303 are on-chip devices, thus inaccessible by a user, which is ideal for ensuring important, permanent and proprietary code cannot be altered by a user. The first processor 305 decodes the frames and stores the decoded frames in the DRAM 309.

[0044] The process of decoding and display of the frames can be implemented as firmware executed by one processor while the process for selecting the appropriate frame for display can be implemented as firmware executed by another processor. Because the decoding and display processes are relatively hardware-dependent, the decoding and display processes can be executed in a customized and proprietary processor. The firmware for the decoding and display processes can be implemented in SRAM.

[0045] On the other hand, the process for selecting the frame for display can be implemented as firmware in DRAM that is executed by a more generic, "off-the-shelf" processor, such as, but not limited to, a MIPS processor or a RISC processor. The foregoing is advantageous because by

offloading the firmware for selecting the frame for display from the SRAM, less space on an integrated circuit is consumed. Additionally, empirically, the process for selecting the image for display has been found to consume the greatest amount of time for debugging. By implementing the foregoing as firmware executed by an "off-the-shelf" processor, more debugging tools are available. Accordingly, the amount of time for debugging can be reduced.

[0046] **Fig.** 4 illustrates a dynamic random access memory (DRAM) unit 309, in accordance with an embodiment of the present invention. The DRAM 309 may contain frame buffers 409, 411 and 413 and corresponding parameter buffers or BDSs 403, 405 and 407.

[0047] In one embodiment of the present invention, the video data is provided to the processor 305. The display device 311 sends a vertical synchronization (vsynch) signal every time it is finished displaying a frame. When a vsynch is sent, the processor 305 may decode the next frame in the decoding sequence, which may be different from the display sequence as explained hereinabove. Since the second processor may be an "off-the-shelf" processor, real-time responsiveness of the second processor may not be guaranteed. To allow the second processor 307 more time to select the frame for display, it is preferable that the second processor 307 selects the frame for display at the next vsynch, responsive to the present vsynch. Accordingly, after the vsynch, the first processor 305 loads parameters for the next decoded frame into the BDS. The second processor 307 can determine the next frame for display, by examining the BDS for all of the frame buffers. This decision can be made prior to the decoding of the next decoded frame, thereby allowing the second processor 307 a window of almost one display period prior to the next vsynch

for determining the frame for display, thereat. The decoded frame is then stored in the appropriate buffer.

[0048] The process of displaying the picture selected by the second processor prior to the latest vsynch may also be implemented utilizing the second processor. Consequently, the first processor may not need to interface with the display hardware and may work based only on the vsynchs and the signals for determining which frame to overwrite from the second processor. The processor 307 notifies the processor 305 of the decision regarding which frame should be displayed next. When the display device 311 sends the next vsynch signal, the foregoing is repeated and the processor 305 displays the frame that was determined by processor 307 prior to the latest vsynch signal. The processor 305 gets the frame to display and its BDS from the DRAM 309, applies the appropriate display parameters to the frame, and sends it for display on the display device 311.

[0049] **Fig.** 5 illustrates a timing diagram of the decoding and displaying process, in accordance with an embodiment of the present invention. Processor 307 may be an off-the-shelf generic processor, and it may be off-chip. If the first processor 305 were the only processor in the system, the code would only be in SRAM 303, and the processor 305 would be able to process, i.e. decode an image and determine the next image to display during a short period after the vsynch, then display that image during the time before the next vsynch signal. However, that is not desirable in this system since, as mentioned hereinabove, utilizing only the on-chip processor 305 and the SRAM 303 may be more costly than utilizing a second processor 307 and a DRAM 309. The second processor 307 and the DRAM 309 may be more economical.

[0050] When the vsynch occurs the processor 307 determines the appropriate image to display next according to the correct order. It is desirable to provide processor 307 with the necessary information to determine the next frame for display early, to afford the processor 307 more time to determine the next frame for display. As a result, when, for instance, a vsynch 0 occurs, processor 305 and processor 307 are notified. Processor 305 may then load the BDS information associated with the frame that is being currently decoded onto the DRAM 309 and notifies the processor 307. The processor 307 determines the appropriate frame to display next to maintain the correct order of frames based on the BDS. Meanwhile, the processor 305 also begins displaying the current display frame, e.g. frame 0, and decoding the current decode frame utilizing instructions stored on the SRAM 303. Once the decoding of the current frame is complete, the processor 305 may send the decoded frame to the DRAM 309. Once the processor 307 determines, for instance, frame 1 that needs to be displayed next, processor 307 sends the determination information to processor 305. At vsynch 1, the foregoing is repeated and processor 305 displays the frame selected after vsynch 0, e.g. frame 1, from the DRAM 309.

[0051] The embodiments described herein may be implemented as a board level product, as a single chip, application specific integrated circuit (ASIC), or with varying levels of the decoder system integrated with other portions of the system as separate components. The degree of integration of the decoder system will primarily be determined by the speed and cost considerations. Because of the sophisticated nature of modern processor, it is possible to utilize a commercially available processor, which may be implemented external to an ASIC implementation. Alternatively, if the processor is available as an ASIC core

or logic block, then the commercially available processor can be implemented as part of an ASIC device wherein certain functions can be implemented in firmware.

[0052] While the present invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the present invention. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the present invention without departing from its scope. Therefore, it is intended that the present invention not be limited to the particular embodiment disclosed, but that the present invention will include all embodiments falling within the scope of the appended claims.